International Journal of Scientific Engineering and Applied Science (IJSEAS) - Volume-1, Issue-4, June 2015
ISSN: 2395-3470
www.ijseas.com

A Novel Approach for Identification of Karaka Relations using Semantic Role Labeling Method

Amita[1], Ajay Jangra[1]
[1] University Institute of Engineering and Technology, Kurukshetra University, Kurukshetra, India

Abstract
An NLP system consists of a grammar and procedural components. The grammar is used by the procedural components in performing analysis, generation, etc. In our work, we also talked about how we can extract the grammatical information. This information includes karaka relations. For this purpose, we find out the issues that come while finding karaka relations and then resolve them with semantic role labeling method. To identify karaka relations in English is very difficult task due to its fixed word order. This paper describes the semantic role labeling method to resolve the issues for identifying karaka relations.
Keywords: *Paninian Grammar, Parser, Semantic role labeling, word order.*

1. Introduction

In Paninian grammar framework, when a structure is formed by grouping of two or more words. We call this structure as karaka relations. There are basically six types of Karaka relations.

- k1 : central to the action of the verb
- k2: the one most desired bby the karta
- k3: instrument which is essential for the action to take place
- k4: recipient of the action
- k5: movement away from the source
- k7: location of the action

Karaka relations are syntactico-semantic (or semantico-syntactic) relations between the verbs and other related constituents (typically nouns) in a sentence [7]. For this we are finding the karaka relations using the paninian grammar. Verb is the central part of the sentence. By using the semantic role method, we are finding the semantics of the sentence and hence the karaka relations. With this information of karaka relations, we can create a dependency treebank for English language.

In the following sections, first we have discussed the annotation scheme for English and then we presented our approach for finding the karaka relations for English sentences based on the Paninian grammar framework.

2. Related work

Uma Maheshwar Rao G. et al., in 2010 [1] attempted to develop a morphological analyzer and generators for South Dravidian languages. A network and process model was developed by K. Narayana Murthy in 2001 [2] for Kannada morphological analysis/ generation name MORPH and on general text the system gives performance of 60 to 70%. For Bengali, unsupervised methodology was used in developing a Morphological Analyzer system and two-level morphology approach was used to handle Bengali compound words. Rule based Morphological Analyzer was developed for Sanskrit.
H. Chaudhary et al. in 2011 [7] discussed an application of the CPG to English and reported how it can account for English, a fixed word order language [8]. And also described the handling of some English constructions and the extensions and adaptations that were made in the Annotation tag set. Further, also discussed the issues encountered during annotation, along with their resolution. Annotation schemes for 'be' verbs, idioms, expletive subjects, 'But' involving sentences also described. For example,
He had to agree in the face of her strong determination.
Though 'in the face of' appears as an idiom in English dictionaries, in order to annotate this sentence as per the HyDT Scheme, the authors chunk the clause 'in the face of her determination' as 'in the face' and 'of her strong determination' (see Figure 1).

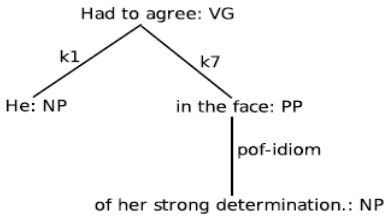

Figure 1. Annotation of Idioms

This helps retain the HyDT framework grammar while annotating this type of sentences. Further, the authors mark the relation between these two chunks as 'pof-idiom' so as to retain the information that 'in the face of' was an idiom, and is followed by an NP. However, dependency annotation of idioms was currently a grey area due to their idiosyncratic behavior, and it needs to be looked into further, as to the level at which they should be captured in the HYDT annotation scheme, and whether they require introduction of another layer of annotation to recognize and mark them as idioms. Especially in cases where they are analyzed more at the syntactic level, based on their grammar.

Gandhe et al., in 2011 [3] considered Indian languages which are considered as low density languages as they do not have rich knowledge sources such as parsers or complex morphological analyzers. These languages also suffer from data sparsity and hence form ideal languages for the analysis of their proposed method. We also consider only various forms of verbs and do not consider other words such as noun phrases and adjectives affected by inflections.

K.Ramakanthan et al., in 2011 [4] used finite state transducer (FST) to develop rule based morphological analyzers (MAG) for Kannada language. The proposed analyzer was capable to generate and analyze list of twenty thousand nouns, around three thousands verbs and smaller list of adjectives. The capacity of system to generate and analyze transitive, causative forms from the passive construction and verbal nouns is the uniqueness of the proposed system. The MAG was developed as part of the machine translation system which worked for English to Kannada language.

M. Parakh and Rajesha N. in 2011 [5] developed a prototype morphological analyzer for four Indian languages. These four languages were Assamese, Bengali, Bodo and Oriya language. The developed prototype model could handles inflectional suffixes and derivation as well as prefixation.

R.Socher et al., in 2012 [6] introduced a Compositional Vector Grammar (CVG), by combining CFGs with a syntactically untied recursive neural networks. The Compositional Vector Grammar improved the Context Free Grammar of the Stanford Parser by 3.8% to obtain 90.4% F1 score. The training and implementation is approximately faster as an efficient re-ranker. The new parser was nearly 20% faster than the current Stanford factored parser. A soft notion of head words is learned by the CVG which improves the performance on the types of ambiguities that require semantic information.

Yue Zhang and Stephen Clark in 2012 [9] developed a dependency parser which is based on graph and transition parser. They combined graph-based and transition based parsing into a single system and proposed a beam searched based parser for training and decoding. They showed that it outperforms both the pure transition-based and the pure graph-based parsers. The accuracy of 92.1% and 86.2% is given by the system byss testing them on English and Chinese Penn Treebank data respectively.

R. N. patel et al., in 2013 [10] proposed a rich set of rules for better reordering. The idea was to facilitate the training process by better alignments and parallel phrase extraction for a phrase based SMT system. Reordering also helped the decoding process and hence improving the machine translation quality. They had observed significant improvements in the translation quality by using our approach over the baseline SMT. They had used BLEU, NIST, multi-reference word error rate, multi-reference position independent error rate for judging the improvements. They had exploited open source SMT toolkit MOSES to develop the system.

H. Chaudhary et al., in 2013 [11] presented the divergence between the treebanks of English and Hindi. The two treebanks diverge mainly from two aspects:

- Stylistic
- Structural

The two treebanks were considered 'divergent' if the parallel trees fell under any of the following:

- Differences in the construction (structure)
- Difference in relations marked (on the parallel sentences)
- Difference in tree depth

- Difference in the frequency of annotation labels

Changes in lexical category of a word of one language and its counterpart in the other, lead to Categorical divergence visible in the data. 'It suffices.' would be translated in Hindi as 'yaha kAfI hE.' (It sufficient is). While the word 'suffices' was realized as the main verb in English it is an adjectival modifier 'kAfI' (sufficient) in the phrase 'kAfI hE', in Hindi. Figure 2 shows the divergent trees for the sentence pair.

Hindi: 'yaha kAfI hE.'
Yaha Kaafii hE
It sufficient is
English: 'It suffices'

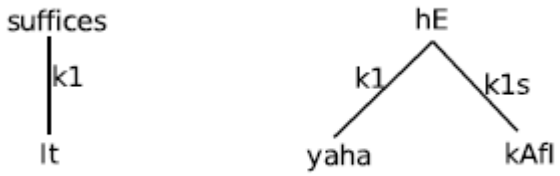

Figure 2. Example showing categorical divergence

We have also done some work related to karaka relations. In this work, we have discussed some background of dependency parsing and issues regarding the karaka relations [12].

3. Existing Problems

The problem while applying paninian framework to English is of mapping. There is no such system that can do exact mapping between English and Hindi dependencies. Stanford parser typed dependencies can be mapped to karaka relations. The various issues that occur are:

- Not exact mapping

Stanford parser parses the English sentences and output the dependencies between tokens. We cannot map these English dependencies to Hindi dependencies directly because of free word order of Hindi language.

- Copula verbs

Copula verbs show the relation between the copular verb and the complement of a copular verb [1]. (We normally take a copula as a dependent of its complement)

"Bill is big" cop(big, is)
"Bill is an honest man" cop(man, is)

But there is no concept of copula verbs in Hindi language.

- Control verbs

Control verbs are difficult to handle. Control verbs can change the meaning of sentence. An example of two control verbs promise and persuade is shown in fig.3 and fig.4.

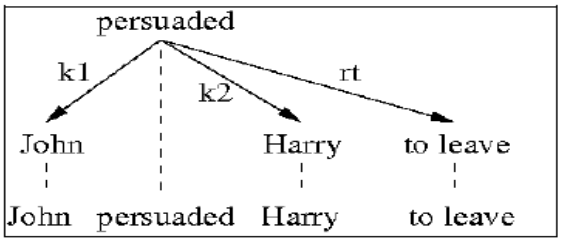

Figure 3. Object control verb

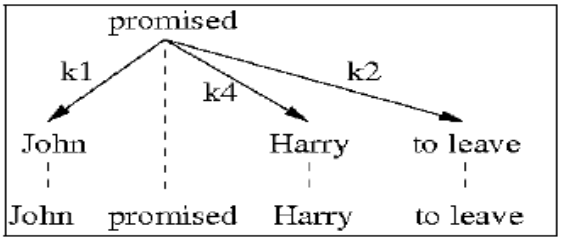

Figure 4. Subject control verb

- Structural differences

There are Structural differences between Hindi and English treebanks.

4. Proposed Solution

Our approach for mapping Stanford typed dependencies is rule based i.e. Manually English sentences output of parser will be study and rules will be created. For semantic role labeling we are using VerbNet. The steps are below:

Step 1: Create simple rules

In this step mapping of dependencies will be done directly, all cases of each token are taken that can occur in sentence like nsubj can be mapped to karta or karma or any other. All cases are written, and in next steps only one by one is removed and in end only one dependency will remain. These direct

mapping is based on the Stanford dependency parser output.

Step 2: Use Verbnet to find verb classes

VerbNet is a lexicon of Approximately 5800 English verbs, and groups Verbs according To shared Syntactic behaviors, Thereby revealing Generalizations of verb behavior. VerbNet is a domain--independent verb lexicon consisting of over 270 such Verb classes

- Extract Lemma of Verb
- Find the Verb class
- Identify Semantic roles and frames
- Craft rules/constraints specific to Verbs.

Step 3: Verb preposition rule

Step 4: Apply rules based on semantic roles & verb class. Semantic roles and thematic roles are also provided by the VerbNet, with roles control verbs can be handled in sentences.

Step 5: Eliminate roles till only one karaka relation per tag or new rule can e applied. In the end only one dependency should be remain so for this, extra roles will be cut down or new can be created.

Let us have an example to understand all the above steps.

- Jessica loaded the boxes into the wagon

The parsed tree by the Stanford parser is:

```
(ROOT
  (S
    (NP (NNP Jessica))
    (VP (VBD loaded)
      (NP (DT the) (NNS boxes))
      (PP (IN into)
        (NP (DT the) (NN wagon))))))
```

The frame given by the VerbNet is:

- <SYNTAX>
- <NP value="Pivot">
<SYNRESTRS />
</NP>
<VERB />
- <NP value="Theme">
<SYNRESTRS />
</NP>
- <PREP value="into">
<SELRESTRS />
</PREP>
- <NP value="location">
<SYNRESTRS />
</NP>
</SYNTAX>

From the above parsed text , we conclude the following information.

NP- pivot – Jessica - k1
V-verb – load – root
NP- theme – boxes – k2
PREP- into
NP – location- wagon – k7

5. Result analysis

For the validation of our results we are using Hindi full parser. Table 1 shows the results for each of the karaka relation separately. First, we count the total number of k1 sentences in the data set and then count the sentences that are matched correctly. On basis of these counts, we find the correctly matched percentage.

TABLE I. TABLE STYLES

Karaka relations	Correct Matching
K1	76.7%
K2	67.7%
K3	79.2%
K4	54.7%
K5	61.6%
K7	81.1%

4. Conclusions

In our work, we have discussed an annotation scheme for English using the Paninian grammar framework. For this purpose, first we have identified the issues encountered. To resolve these issues, we have presented an approach with semantic role labeling method. Further, our approach can be used to create a dependency treebank for English language.

References

[1] Uma Parameshwari Rao G and Parameshwari K, "*On the description of morphological data for morphological analyzers and generators: A case of Telugu, Tamil and Kannada*", In proceedings of Third linguistics annotation workshop, University of Hyderabad,2010.

[2] K.Narayana Murthy, B.R Shambhavi, B.J Jyothi, and Varsha shastri, "*Kannada Morphological Analyzer And Generator*", in proceeding of International Journal of Computer Science and Network Security (IJCSNS) vol.11 January 2011.

[3] Gandhe, Ankur, Rashmi Gangadharaiah, Karthik Visweswariah, and Ananthakrishnan Ramanathan, "*Handling verb phrase morphology in highly inflected Indian languages for Machine Translation*", IJCNLP, 2011.

[4] K.Ramakanthan, B.R Shambhavi, B.J Jyothi, and Varsha shastri, "*Kannada Morphological Analyzer And Generator*", in proceeding of International Journal of Computer Science and Network Security (IJCSNS) vol.11 January 2011.

[5] M. Parakh and Rajesha N., "*Developing Morphological Analyzer for four Indian languages Using a rule based Affix Stripping approach*" ,Linguistic Data Consortium for Indian Languages, CIIL, Mysore, 2011.

[6] Richard Socher, John Bauer, Christopher D Manning and Andrew Y Ng, " Parsing with compositional vector grammars" , In ACL, 2012.

[7] H. Chaudhary, H. Sharma and D.M. Sharma, "*Annotation and issues in English dependency tree bank*",in proceedings of the Second International Conference on Dependency Linguistics (DepLing 2013), prague, pages 133–140, 2011.

[8] M. Marcus, B. Santorini, and M.A. Marcinkiewicz, "*Building a large annotated corpus of English: The Penn Treebank*", Computational Linguistics 1993.

[9] Y. Zhang and S. Clark "*A tale of two parsers: investigating and combining graph based and transition-based dependency parsing using beam-search*", In Proceedings of Empirical method of natural language processing (EMNLP),Hawaii, USA, 2008

[10] Raj Nath Patel, Rohit Gupta, Prakash B. Pimpale and Sasikumar M., "Reordering rules for English- Hindi SMT", Proceedings of the Second Workshop on Hybrid Approaches to Translation. Association for Computational Linguistics, 2013.

[11] H. Chaudhary, H. Sharma and D.M. Sharma, "*Divergences in English-Hindi Parallel Dependency Treebank*",in proceedings of the Second International Conference on Dependency Linguistics (DepLing 2013), prague, pages 33–40, August 27–30, 2013.

[12] Amita and A. Jangra, "*An Annotation scheme for English Language using Paninian Framwork*", in International journal of innovative science, engineering and technology(IJISET), vol. 2, issue 1, January 2015.